

1 REMARKS

2 Status of the Claims

3 Claims 1-23 and 36-58 are pending in the present application, Claims 24-35 having been
4 canceled in the present amendment as being directed to non-elected claims. New Claims 59-61 have
5 been added that clearly define the invention and distinguish over the cited art.

6 Telephone Restriction

7 On June 15, 2005, Examiner Ali issued a telephone restriction, and applicants' attorney
8 (Michael King, Registration No. 44,832) elected Claims 1-23 and 36-57 without traverse, subject to
9 the right to file a divisional application directed to the non-elected claims.

10 Problem Solved by the Present Invention

11 Before discussing the substantive rejection of the claims, it will be useful to briefly
12 articulate at least one of the significant problems the present invention was intended to address. It
13 should be recognized that the cited art does not relate to this problem and thus, is not particularly
14 relevant.

15 It will be appreciated that software applications are frequently revised, particularly during a
16 development phase. Such revisions add new capabilities, add new functionalities, and/or correct
17 errors (i.e., bugs) discovered in earlier versions. In the past, server-hosted software applications have
18 been revised by taking the server-hosted software application off-line (thereby preventing clients
19 from accessing the application), updating the software application, migrating all data associated with
20 the software application to the new version of the software application, and restarting the software
21 application (i.e., enabling clients to access the updated application). This technique is described in
22 the Background of the Invention section of applicants' specification.

23 The present invention provides an alternative approach for updating a server-hosted software
24 application. Essentially, in the present invention, instead of taking a previous version of a software
25 application off-line, a different version of the same software application is loaded onto one of the
26 network servers. This technique is thus much less disruptive than the conventional approach, because
27 a version of the software application is always online and available for client use, and because
28 initially, only a first few clients who are enabled to access the different version of the software
29 application might be affected, if the new version of the software application fails to perform properly
30 or causes undesired or unexpected results.

1 Claims Rejected under 35 U.S.C. § 103

2 The Examiner has rejected Claims 1-23 and 36-58 as being obvious over Peterson
3 (USPGPUB 2002/0103907) in view of Pezutti (USPGPUB 2004/0249927). The Examiner asserts
4 that Peterson discloses most of the elements of the claimed invention, and that Pezutti discloses each
5 remaining element that is not disclosed by Peterson. The Examiner asserts that it would be obvious
6 to one of ordinary skill in the art of data processing to combine the teachings of the cited art, because
7 such a combination would have provided network access services for the benefit of network
8 providers, service providers, and customers. The Examiner further notes that Pezutti suggests that
9 installation of his invention improves service to customers, network providers and service providers,
10 which provides motivation to modify Peterson in view of Pezutti to achieve applicants' claimed
11 approach. Applicants respectfully disagree with this conclusion for the following reasons.

12 The cited art simply does not teach a network environment in which a plurality of different
13 versions of the same software application are simultaneously installed on one or more network
14 servers. The *Free Online Dictionary of Computing* (FOLDOC), a reference with which those of
15 ordinary skill in the computing arts are likely to be acquainted, provides the following definition for
16 the term *version*:

17 One of a sequence of copies of a program, each incorporating new
18 modifications. Each version is usually identified by a number, commonly of
19 the form X.Y where X is the major version number and Y is the release
20 number. Typically an increment in X (with Y reset to zero) signifies a
21 substantial increase in the function of the program or a partial or total re-
22 implementation, whereas Y increases each time the program is changed in any
23 way and re-released.

24 Version numbers are useful so that the user can know if the program has
25 changed (bugs have been fixed or new functions added) since he obtained his
26 copy and the programmer can tell if a bug report relates to the current version.
It is thus always important to state the version when reporting bugs. Statements
about compatibility between different software components should always say
which versions they apply to (1997-12-07).

27 Significantly the *FOLDOC* entry for the term *version* appears to date from 1997, well before the
28 filing of the present application. It appears that at least as early as 1997, one of ordinary skill in the
29 computer arts would have recognized that the term *version* refers to a unique release of a specific software
30 application, differing from previous releases either by offering some additional functionality or capability,

1 or correcting an error or other issue (such as a bug) present in an earlier release of the software
2 application. Significantly, different versions are simply not different copies of the same software
3 applications. Instead, different versions incorporate some type of code change that offers additional
4 functionality or corrects an error or bug found in an earlier version of the software application.

5 It is also noteworthy that the USPTO Classification relating to data processing (Developing,
6 Installation and Management, Class 717) specifically recognizes the term *version* in the context of
7 code changes to a software application.

8 Subject matter including means or steps for maintaining different versions of
9 source code under development in a library to facilitate software development.

10 (1) Note. For the purpose of this definition, the versions may have been
11 created by a plurality of programmers and/or at different times.

12 (2) Note. For the purpose of this definition, examples of source code
13 version management include UNIX utilities Source Code Control System
14 (SCCS) and Revision Control System (RCS) (*USPTO Patent Classification,*
15 *Class Definitions, Class 717, Section II, References to Other Classes,*
Class 122).

16 Thus, the Patent Office itself appears to recognize that in the context of the computer arts, the
17 term *version* has a well understood meaning with respect to software applications.

18 There does not appear to be any basis for concluding that the plain meaning of the term *multiple*
19 *versions* could have been interpreted by one of ordinary skill in the art at the time of the invention as
20 being anything other than different releases of the same software application, where there is a difference
21 in code (i.e., for the correction of a bug or the incorporation of some additional functionality or capability)
22 between the different versions. Indeed, the specification as filed clearly describes the development
23 process of software that results in the release of successive versions of the same software application.

24 Clearly, one of ordinary skill in the art at the time of the invention would have recognized that the
25 term *multiple versions* of a server-hosted application meant something more than *multiple copies* of a
26 server-hosted application. Thus, in the context of the present invention, the network must include
27 multiple different versions or releases of the same server-hosted application, which is different than
28 simply including multiple copies (with the same code) of the server-hosted application.

29 Turning now to the Peterson reference and what is disclosed by Peterson with respect to
30 updating, it will be apparent that Peterson does not teach or suggest a network environment

1 incorporating multiple versions (i.e., different releases) of a software application. While the art cited
2 by the Examiner clearly involves a networked environment, it does not appear that the cited art addresses
3 the problem addressed by the present invention, i.e., the problem of efficiently and smoothly migrating
4 clients from one version of a server-hosted application to another version of the server-hosted application.
5 Significantly, Peterson is directed to solving a different problem, i.e., allocating storage available on a
6 plurality of networked servers so that clients can make efficient use of the storage resources over the
7 network. While it is true that Peterson teaches that each server with storage capacity is executing the
8 same application (the Internet File Client application), Peterson does not teach or suggest that any of those
9 multiple copies are *different versions* of that software application. Neither Peterson, nor any of the other
10 cited art teaches or suggests updating a version of a server-hosted application, such that on a network, at
11 least two different versions of the same server-hosted application are concurrently in use. The provision
12 of two different versions of the same server-hosted application that are simultaneously usable is a key
13 advantage of applicants' claimed approach that is not taught or suggested by the cited art.

14 Pezutti appears to be related to a network architecture including customers, service retailers,
15 service networks, and access networks and discusses networks that appear to be particularly well-suited to
16 telecommunications. Significantly, it does not appear that Pezutti teaches or suggests any techniques
17 useful to facilitate the migration of clients from one version of a server-hosted software application to a
18 different version of the server-hosted software application in the manner recited by applicants' claims.

19 It appears that the Examiner is interpreting the term *version* to have the same meaning as *copy*, such
20 that the *multiple copies* of the Internet File Client application disclosed by Peterson (paragraph 67) read on
21 the *multiple versions* recited by applicants' claims. Or, the Examiner may have concluded that the FAT
22 server program implemented on a central server, the Internet File Server program implemented on a client
23 (i.e.; sever 110), and the Internet File Client applications implemented on the remote storage devices (i.e.,
24 the network locations where the client's data will be stored) each represents a *different version* of a server
25 hosted application. It is important to recognize that both possible interpretations are inconsistent with the
26 accepted definition/plain meaning of term *version*, as discussed above. MPEP 2111.01 clearly indicates that
27 claim terms are presumed to have the ordinary and customary meanings attributed to them by those of
28 ordinary skill in the art. Applicants have submitted evidence that the term *version* has a specific meaning
29 that would be recognized by one of ordinary skill in the art, which is not consistent with either interpretation
30 that the Examiner might be applying. With respect to Peterson's disclosure, there is simply no evidence that

1 the plurality of copies of the Internet File Client (one copy installed on each storage device) represent
2 *different versions* of the Internet File Client software application. Furthermore, there is no evidence that the
3 FAT server program, the Internet File Server program, and the Internet File Client applications disclosed by
4 Peterson correspond to *different versions* of the software application, but are instead, different software
5 applications altogether. When applicants' claims are analyzed using the term *version* consistent with the
6 plain meaning of the term as would be recognized by one of ordinary skill in the art, applicants' claims are
7 readily distinguishable over the cited art.

8 Because Peterson does not address or relate to the problem of how to update a server-hosted
9 software application from a first version to a second version on the network, without adversely impacting
10 clients, and because Pezutti similarly does not address the issue of migrating clients from one version of a
11 server-hosted software application to another version of the server-hosted software application, there is no
12 reason why one of ordinary skill in the art would be led to applicants' claimed invention by combining
13 Peterson and Pezutti. These two references do not teach or suggest applicants' recitation in the claims of
14 this application. Significantly, neither reference relates to a network implementing multiple versions of
15 the same server hosted application. Accordingly, the rejection of Claims 1-23 and 36-58 as being obvious
16 over Peterson in view of Pezutti should be withdrawn.

17 The preceding comments apply to each independent claim. It should be noted that applicants
18 have not specifically discussed the patentability of each dependent claim over the cited art, and have
19 chosen to forgo such an analysis in an effort to reduce the complexity and length of this response.
20 That decision should not be construed as an admission that the dependent claims are not patentably
21 distinguished over the cited art for additional reasons.

22 Claims 1, 18, 36 and 58 further distinguish over the cited art, because these claims recite a
23 register that associates each client with a specific *version* of the server-hosted application, and
24 changing that register each time a client is migrated from one *version* to another. The cited art does
25 not teach or suggest simultaneously hosting multiple *versions* of the same server-hosted application,
26 or any equivalent element. Claims 1, 18, 36 and 58 are thus distinguishable over the cited art for this
27 additional reason.

28 Patentability of Newly Added Claims

29 New Claims 59, 60, and 61 have been added in the present amendment. None of these claims
30 introduce new matter, and none of these claims raises a new issue requiring a further search. These

1 new claims simply rewrite existing independent claims, including additional the language specifically
2 defining the term *version*, consistent with the use of the term in the specification as filed, and the
3 plain meaning of the term as would be recognized by one of ordinary skill in the art.


4 Claim 59 is based on Claim 1, and further recites: "*the multiple versions including at least*
5 *two different releases of same the server-hosted application.*" As discussed in detail above, the cited
6 art does not teach or suggest a networked environment in which a plurality of different versions of the
7 same server hosted application are implemented.

8 Similarly, Claim 60 is based on Claim 18, and further recites: "*the multiple versions*
9 *including at least two different releases of same the server-hosted application.*" As discussed in
10 detail above, the cited art does not teach or suggest a networked environment in which a plurality of
11 different versions of the same server hosted application are implemented.

12 Claim 61 is based on Claim 36, and further recites: "*the first version and the second version*
13 *corresponding to two different releases of same the server-hosted application.*" Again, the cited art does
14 not teach or suggest a network implementing different versions of the same server hosted application.

15 In view of the amendments and the Remarks set forth above, it will be apparent that the claims in
16 this application define a novel and non-obvious invention, and that the application is in condition for
17 allowance and should be passed to issue without further delay. Should any further questions remain, the
18 Examiner is invited to telephone applicants' attorney at the number listed below.

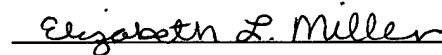
19 Respectfully submitted,

20 
21
22 Michael C. King
23 Registration No. 44,832

24 MCK:klp

25 I hereby certify that this correspondence is being deposited with the U.S. Postal Service in a sealed
26 envelope as first class mail with postage thereon fully prepaid addressed to: Commissioner for Patents,
27 Alexandria, VA 22313-1450, on September 23, 2005.

28 Date: September 23, 2005

29 
30